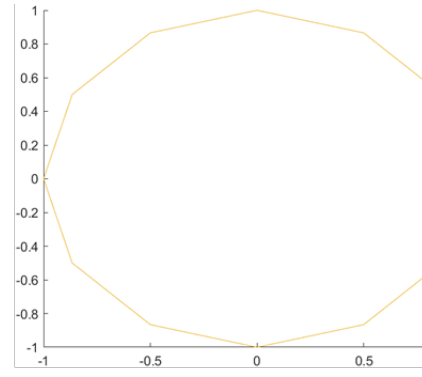
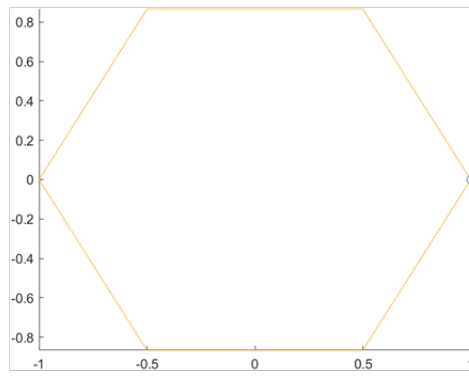
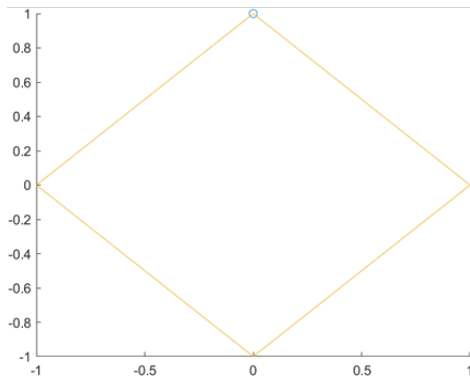


Actividad 2.2 (Parametrización de trayectorias)

Daniel Castillo López A01737357

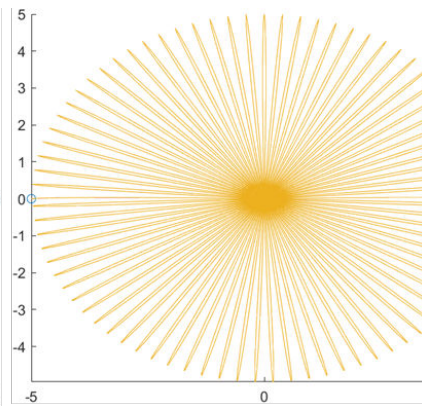
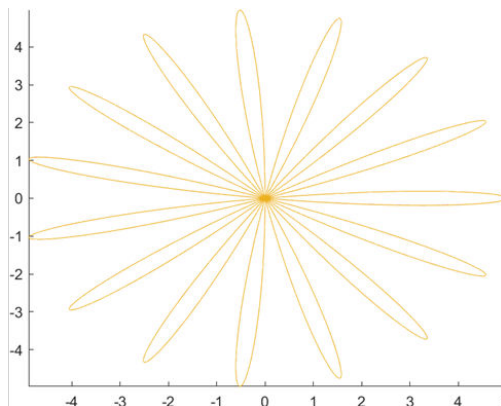
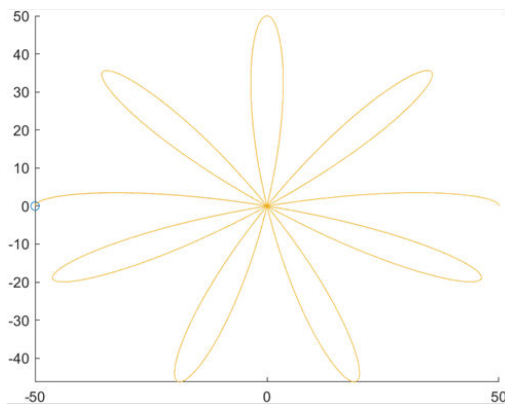
1. **Implementar** el código requerido para generar la parametrización de las siguientes trayectorias en un plano 2D.

2.



3. **Implementar** el código requerido para generar la parametrización de las siguientes trayectorias en un plano 2D.

4.



Cuadrado

Creamos un vector llamado tiempo, que toma los valores del 1 al 6 en un incremento de 1

```
%Se define el parámetro "t" de parametrización sobre el cual se realizará  
%la proyección de trayectoria  
tiempo=[1:1:6];
```

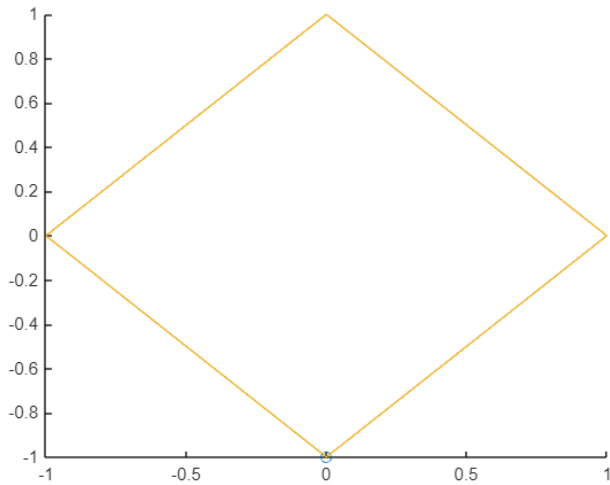
Al normalizar el vector del tiempo par que sus valores se ajusten al rango de π y 3.5π , transformando que el mínimo corresponda a π y su máximo sea $3.5 * \pi$

```
%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo  
%theta  
t1= normalize(tiempo,"range",[pi,3.5*pi]);
```

Cuando creamos una matriz que vaya recopilando las posiciones de

```
%Obtenemos las ecuaciones dependientes del tiempo  
x1= ones(1,6).*cos(t1);  
y1=ones(1,6).*sin(t1);
```

```
%graficamos la trayectoria
figure(1)
comet(x1, y1)
```



Hexágono

Desarrollamos un cambio en cuantos vectores pueden entrar para la figura.

```
%Se define el parámetro "t" de parametrización sobre el cual se realizará
%la proyección de trayectoria
```

```
tiempo=[1:1:7];
```

```
%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
%theta
```

```
t2= normalize(tiempo, "range", [pi, 3*pi]);
```

Modificando de igual manera la matriz que se forma con 7 valores, valores que chocan en la gráfica para formar la figura

```
%Obtenemos las ecuaciones dependientes del tiempo
```

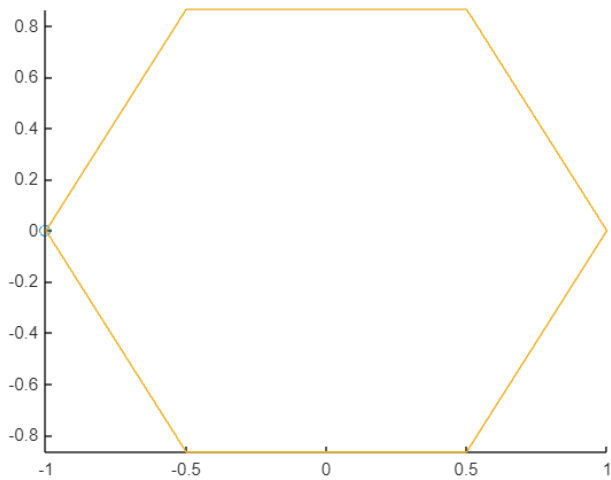
```
x2= ones(1,7).*cos(t2);
```

```
y2= ones(1,7).*sin(t2);
```

```
%graficamos la trayectoria
```

```
figure(2)
```

```
comet(x2, y2)
```



Dodecágono

```
%Se define el parámetro "t" de parametrización sobre el cual se realizará
%la proyección de trayectoria
```

```
tiempo=[1:1:13];
```

```
%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
%theta
```

```
t3= normalize(tiempo,"range",[pi,3*pi]);
```

```
%Obtenemos las ecuaciones dependientes del tiempo
```

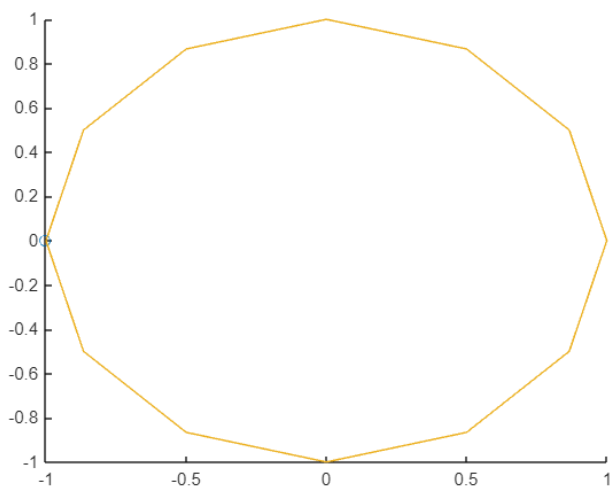
```
x3= ones(1,13).*cos(t3);
```

```
y3= ones(1,13).*sin(t3);
```

```
%graficamos la trayectoria
```

```
figure(3)
```

```
comet(x3, y3)
```



Flor de 7 pétalos

Aquí se crea un vector de tiempo desde 0 hasta 10, con incrementos de 0.01.

Esto genera 1001 puntos para suavizar la curva al graficarla.

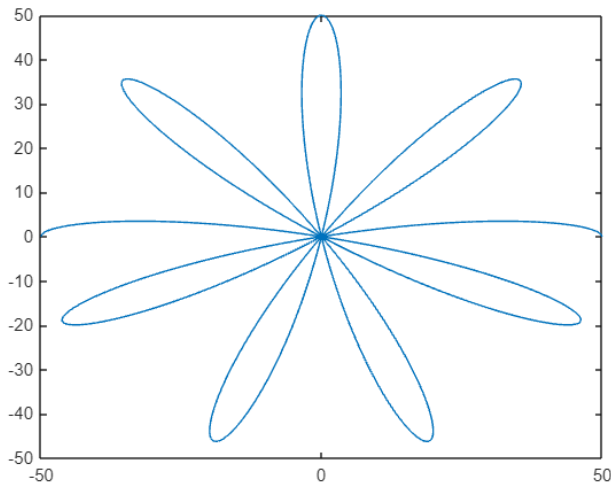
```
%Graficación de la trayectoria en coordenadas paramétricas
```

```
%Se define el parámetro "t" de parametrización sobre el cual se realizará  
%la proyección de trayectoria  
tiempo = [0:0.01:10];
```

```
%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo  
%theta  
t4 = normalize(tiempo, "range", [0, pi]);
```

$\cos(8*t4)$: Modula el radio de la trayectoria, haciendo que se contraiga y se expanda en función de t

```
%Obtenemos las ecuaciones dependientes del tiempo  
x4 = 50*cos(8*t4).*cos(t4);  
y4 = 50*cos(8*t4).*sin(t4);  
%graficamos la trayectoria  
figure(4)  
%comet(x4, y4)  
plot(x4,y4)
```



Flor de 15 pétalos

```
%Graficación de la trayectoria en coordenadas paramétricas
```

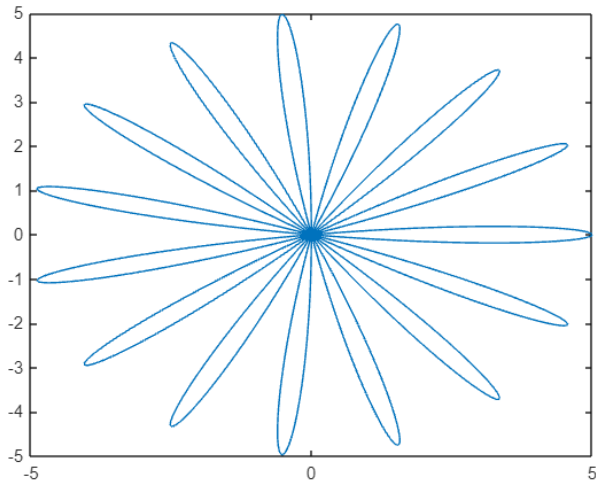
```
%Se define el parámetro "t" de parametrización sobre el cual se realizará  
%la proyección de trayectoria  
tiempo = [0:0.01:10];
```

```

%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
%theta
t5 = normalize(tiempo, "range", [0, pi]);

%Obtenemos las ecuaciones dependientes del tiempo
x5 = 5*cos(15*t5).*cos(t5);
y5 = 5*cos(15*t5).*sin(t5);
%graficamos la trayectoria
figure(5)
%comet(x5, y5)
plot(x5,y5)

```



Flor n petalos

```

%Graficación de la trayectoria en coordenadas paramétricas

%Se define el parámetro "t" de parametrización sobre el cual se realizará
%la proyección de trayectoria
tiempo = [0:0.01:10];

%Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
%theta
t6 = normalize(tiempo, "range", [0, pi]);

%Obtenemos las ecuaciones dependientes del tiempo
x6 = 5*cos(75*t6).*cos(t6);
y6 = 5*cos(75*t6).*sin(t6);
%graficamos la trayectoria
figure(6)
%comet(x6, y6)
plot(x6,y6)

```

